
pymills Documentation

Release 3.4.3.dev

James Mills

March 31, 2016

1	About	3
1.1	Installation	3
2	Documentation	5
2.1	API Documentation	5
2.2	TODO	15
2.3	Road Map	16
2.4	Changes	16
3	Indices and tables	17
	Python Module Index	19

Release 3.4.3.dev

Date March 31, 2016

About

pymills is a collection of various useful [Python](#) utility modules maintained by [James Mills](#). Some are written by James Mills while others are borrowed from other sources and attributed.

Note: You may not find this library/package useful to you as I (*James Mills*) don't actively maintain this as a real project as such. It was my first ever [Python](#) Package (*and the name has stuck*) and I generally just throw things in it that I can find no better place for. See the [API](#) to find anything of interest.

- Visit the [Project Website](#)
- [Read the Docs](#)
- Download it from the [Downloads Page](#)

1.1 Installation

The simplest and recommended way to install pymills is with pip. You may install the latest stable release from PyPI with pip:

```
> pip install pymills
```

If you do not have pip, you may use easy_install:

```
> easy_install pymills
```

Alternatively, you may download the source package from the [PyPI Page](#) or the [Downloads page](#) on the [Project Website](#); extract it and install using:

```
> python setup.py install
```

You can also install the [latest-development version](#) by using pip or easy_install:

```
> pip install pymills==dev
```

or:

```
> easy_install pymills==dev
```

For further information see the [pymills documentation](#).

Documentation

2.1 API Documentation

2.1.1 pymills Package

pymills Package

pymills

Mills Python Library

copyright Copyright (C) 2004-2013 by James Mills

license MIT (See: LICENSE)

datatypes Module

Data Types Library

Various custom data types not available in the standard python library.

class `pymills.datatypes.OrderedDict` (*d={}*)

Bases: `dict`

clear ()

items ()

keys ()

popitem ()

setdefault (*key, failobj=None*)

update (*d*)

iteritems ()

itervalues ()

values ()

iterkeys ()

index (*key*)

```
class pymills.datatypes.Stack (size=None)
    Bases: object

    push (item)

    pop ()

    peek (n=0)

    empty ()

class pymills.datatypes.Queue (size=None)
    Bases: object

    push (item)

    get (n=0, remove=False)

    pop (n=0)

    peek (n=0)

    top ()

    bottom ()

    empty ()

    size ()

    full ()

class pymills.datatypes.CaselessList
    Bases: list

    append (obj)

    remove (value)

class pymills.datatypes.CaselessDict
    Bases: dict

    get (key, default=None)

    has_key (key)

class pymills.datatypes.Null
    Bases: object
```

dbapi Module

DB-API Wrapper Classes

...

```
pymills.dbapi.mysql_session (*args, **kwargs)
pymills.dbapi.oracle_session (*args, **kwargs)
pymills.dbapi.sqlite_session (*args, **kwargs)
pymills.dbapi.create_connection (type, *args, **kwargs)
pymills.dbapi.Connection (type, *args, **kwargs)
exception pymills.dbapi.DriverError (driver, msg)
    Bases: exceptions.Exception
```

exception `pymills.dbapi.ConnectionError` (*driver, e*)

Bases: `exceptions.Exception`

exception `pymills.dbapi.DatabaseError` (*sql, e*)

Bases: `exceptions.Exception`

class `pymills.dbapi.BaseSession` (*cx*)

Bases: `object`

initializes x; see `x.__class__.__doc__` for signature

cursor (*create=False*)

close ()

rollback ()

commit ()

execute (*sql=None, *args, **kwargs*)

do (*sql=None, *args, **kwargs*)

class `pymills.dbapi.SQLiteSession` (*cx*)

Bases: `pymills.dbapi.BaseSession`

initializes x; see `x.__class__.__doc__` for signature

class `pymills.dbapi.MySQLSession` (*cx*)

Bases: `pymills.dbapi.BaseSession`

initializes x; see `x.__class__.__doc__` for signature

class `pymills.dbapi.OracleSession` (**args, **kwargs*)

Bases: `pymills.dbapi.BaseSession`

ORACLE_ARRAYSIZE = 2048

class `pymills.dbapi.Records` (*session, cursor*)

Bases: `object`

initializes x; see `x.__class__.__doc__` for signature

next () → the next value, or raise `StopIteration`

class `pymills.dbapi.Record` (*data=(), **kws*)

Bases: `pymills.pyodict.odict`

This doesn't accept keyword initialization as normal dicts to avoid a trap - inside a function or method the keyword args are accessible only as a dict, without a defined order, so their original order is lost.

emailtools Module

Email Tools

A feature-rich Email class allowing you to create and send multi-part emails as well as a simple `sendEmail` function for simpler plain text emails.

class `pymills.emailtools.Email` (*sender, recipients, subject='', cc=[], bcc=[]*)

Bases: `object`

add (*text='', file=None, attach=False, filename=None*)

send ()

`pymills.emailtools.sendEmail (sender, recipient, subject, message) → None`
Send a simple email to the given recipient with the given subject and message.

`pymills.emailtools.get_mimetype (filename)`

`pymills.emailtools.mimify_file (filename)`

`pymills.emailtools.send_email (to, subject, text, **params)`

mathtools Module

Math Tools

Module of small useful math tools aka common math routines.

`pymills.mathtools.mean (xs)`
Calculate the mean of a list of numbers given by xs

`pymills.mathtools.std (xs)`
Calculate the standard deviation of a list of numbers give by xs

misc Module

Miscellaneous

Various miscellaneous functions that don't fit Use as documented.

`pymills.misc.hmsToSeconds (h=0, m=0, s=0) → int`
Calculates the number of seconds, given hours (h), minutes (m) and seconds (s=0).

Example:

```
>>> hmsToSeconds(1)
3600
>>> hmsToSeconds(1, 24)
5040
>>> hmsToSeconds(1, 24, 54)
5094
```

`pymills.misc.getTotalTime (s, e)`

`pymills.misc.addPercent (value, percentage)`

`pymills.misc.subPercent (value, percentage)`

`pymills.misc.bytes (bytes)`

`pymills.misc.duration (seconds)`

`pymills.misc.backMerge (l, n, t=' ')`
Merge the last n items in list l joining with tokens t

l - list n - merge last n items from l t - token (default: " ")

`pymills.misc.strToBool (s)`

`pymills.misc.beat ()`

`pymills.misc.buildAverage (stime, x)`

pyodict Module

class `pymills.pyodict.odict` (*data=()*, ***kws*)
 Bases: `pymills.pyodict._odict`, `dict`

This doesn't accept keyword initialization as normal dicts to avoid a trap - inside a function or method the keyword args are accessible only as a dict, without a defined order, so their original order is lost.

table Module

Table generation, display and manipulation module.

Classes to generate ASCII and HTML tables and traverse data.

class `pymills.table.Header` (*name*, *table=None*, ***kwargs*)
 Bases: `object`

Create a new Header

Container class that holds the definition of a table header and how each piece of data should be formatted and displayed.

An Optional list of kwargs can also be supplied that affect how this header is created:

- type** - the data type. Must be compatible with `type(x)`
- align** - text alignment. one of "left", "center", or "right"
- hidden** - whether this column of data and header is displayed.
- format** - format str or callable used to format cells
- width** - width to use when formatting strings by `__str__` (used by align)
- cls** - class attribute used by toHTML
- style** - style attribute used by toHTML
- ccls** - cell class attribute to use for each cell used by toHTML
- cstyle** - cell style attribute to use for each cell used by toHTML

Example:

```
>>> h = Header("Test")
>>> print h
Test
>>> print h.toHTML()
<th>Test</th>
>>>
```

initializes x; see `x.__class__.__doc__` for signature

refresh ()

getWidth ()

toHTML ()

class `pymills.table.Row` (*cells*, *table=None*, ***kwargs*)
 Bases: `list`

Create a new Row

Container class that holds the definition of a table row and a set of cells and how each cell should be formatted and displayed.

An Optional list of kwargs can also be supplied that affect how this row is created:

- hidden - whether this row is displayed.
- cls - class attribute used by toHTML
- style - style attribute used by toHTML

Example:

```
>>> c = Cell(22.0/7.0, format="%0.2f", align="right", cls="foo", style="hidden: true")
>>> r = Row([c], cls="asdf", style="qwerty")
>>> print r
3.14
>>> print r.toHTML()
<tr class="asdf" style="qwerty"><td align="right" class="foo" style="hidden: true">3.14</td></tr>
>>>
```

initializes x; see x.__class__.__doc__ for signature

refresh()

toHTML()

class pymills.table.**Cell**(value, **kwargs)

Bases: object

Create a new Cell

Container class that holds the definition of a table cell and it's value and how it should be formatted and displayed.

An Optional list of kwargs can also be supplied that affect how this header is created:

- type - the data type. Must be compatible with type(x)
- align - text alignment. one of "left", "center", or "right"
- hidden - whether this cell is displayed
- format - format str or callable
- cls - class attribute used by toHTML
- style - style attribute used by toHTML

Example:

```
>>> c = Cell(22.0/7.0, format="%0.2f", align="right", cls="foo", style="hidden: true")
>>> print c
3.14
>>> print c.toHTML()
<td align="right" class="foo" style="hidden: true">3.14</td>
>>>
```

initializes x; see x.__class__.__doc__ for signature

refresh()

toHTML()

getHeader()

class `pymills.table.Table` (*headers*, *rows*=[], ***kwargs*)

Bases: `list`

Create a new Table

Container class to hold a set of rows and headers allowing easy traversal and display.

An Optional list of kwargs can also be supplied that affect how this table is created:

- `cls` - class attribute used by `toHTML`
- `style` - style attribute used by `toHTML`

Example:

```
>>> c = Cell(22.0/7.0, format="%0.2f", align="right", cls="foo", style="hidden: true")
>>> r = Row([c], cls="asdf", style="qwerty")
>>> h = Header("Test")
>>> t = Table([h], [r])
>>> print t
Test
----
3.14
----

>>> print t.toHTML()
<table><th>Test</th><tr class="asdf" style="qwerty"><td align="right" class="foo" style="hidden: true">3.14</td></tr></table>
>>>
```

initializes `x`; see `x.__class__.__doc__` for signature

append (*row*)

refresh ()

toHTML ()

getXY (*x*, *y*)

`pymills.table.test` ()

utils Module

Utilities

Various utility classes and functions.

exception `pymills.utils.Error`

Bases: `exceptions.Exception`

Error Exception

class `pymills.utils.State`

Bases: `object`

Create new State object

Creates a new state object that is suitable for holding different states of an application. Useful in state-machines.

The way this works is rather simple. You create a new state object, and simply set the state. If the state doesn't exist, it's added to its internal data structure. The reason this is done is so that comparing states is consistent, and you can't just compare with a non-existent state.

initializes `x`; see `x.__class__.__doc__` for signature

set (*s*) → None

Set the current state to the specified state given by *s*, adding it if it doesn't exist.

`pymills.utils.getFiles` (*root*, *pattern*=".*", *tests*=[*isfile*], ***kwargs*) → list of files

Return a list of files in the specified path (*root*) applying the predicates listed in *tests* returning only the files that match the pattern. Some optional *kwargs* can be specified:

- **full**=True (Return full paths)
- **recursive**=True (Recursive mode)

`pymills.utils.isReadable` (*file*) → bool

Return True if the specified file is readable, False otherwise.

`pymills.utils.mkpasswd` (*length*=8, *digits*=2, *upper*=2, *lower*=2)

Create a random password

Create a random password with the specified length and no. of digit, upper and lower case letters.

Parameters

- **length** (*int*) – Maximum no. of characters in the password
- **digits** (*int*) – Minimum no. of digits in the password
- **upper** (*int*) – Minimum no. of upper case letters in the password
- **lower** (*int*) – Minimum no. of lower case letters in the password

Returns A random password with the above constraints

Return type str

`pymills.utils.validateEmail` (*email*) → bool

Return True if the specified email is valid, False otherwise.


```
pymills.utils.safe_import_ (moduleName, globals={ 'getFiles': <function getFiles
at 0x7f27e78586e0>, 'chain': <type 'itertools.chain'>,
'resident': <function resident at 0x7f27e77e0320>,
'Cache': <class 'pymills.utils.Cache'>, 'sample': <bound
method Random.sample of <random.Random object at
0x1ed6d50>>, 'State': <class 'pymills.utils.State'>, 'seed':
<bound method Random.seed of <random.Random ob-
ject at 0x1ed6d50>>, 'minmax': <function minmax at
0x7f27e77e05f0>, 'isfile': <function isfile at 0x7f27ecadd0c8>,
'safe_import_': <function safe_import_ at 0x7f27e7858cf8>,
'__package__': 'pymills', 're': <module 're' from
'/home/docs/checkouts/readthedocs.org/user_builds/pymills/envs/latest/lib/python2.7/re.pyc'>,
'memory': <function memory at 0x7f27e7858f50>, 'stacksize':
<function stacksize at 0x7f27e77e0398>, '__doc__': 'Util-
ities\n\nVarious utility classes and functions.\n', 'printdict':
<function printdict at 0x7f27e77e0488>, 'string': <module 'string'
from '/usr/lib/python2.7/string.pyc'>, '__builtins__': { 'bytearray':
<type 'bytearray'>, 'IndexError': <type 'exceptions.IndexError'>,
'all': <built-in function all>, 'help': Type help() for interactive
help, or help(object) for help about object., 'vars': <built-in
function vars>, 'SyntaxError': <type 'exceptions.SyntaxError'>,
'unicode': <type 'unicode'>, 'UnicodeDecodeError': <type
'exceptions.UnicodeDecodeError'>, 'memoryview': <type 'memo-
ryview'>, 'isinstance': <built-in function isinstance>, 'copyright':
Copyright (c) 2001-2014 Python Software Foundation. All Rights
Reserved. Copyright (c) 2000 BeOpen.com. All Rights Reserved.
Copyright (c) 1995-2001 Corporation for National Research Initia-
tives. All Rights Reserved. Copyright (c) 1991-1995 Stichting Math-
ematisch Centrum, Amsterdam. All Rights Reserved., 'NameError':
<type 'exceptions.NameError'>, 'BytesWarning': <type 'excep-
tions.BytesWarning'>, 'dict': <type 'dict'>, 'input': <built-in
function input>, 'oct': <built-in function oct>, 'bin': <built-in
function bin>, 'SystemExit': <type 'exceptions.SystemExit'>,
'StandardError': <type 'exceptions.StandardError'>, 'format':
<built-in function format>, 'repr': <built-in function repr>,
'sorted': <built-in function sorted>, 'False': False, 'Runtime-
Warning': <type 'exceptions.RuntimeWarning'>, 'list': <type
'list'>, 'iter': <built-in function iter>, 'reload': <built-in function
reload>, 'Warning': <type 'exceptions.Warning'>, '__package__':
None, 'round': <built-in function round>, 'dir': <built-in function
dir>, 'cmp': <built-in function cmp>, 'set': <type 'set'>, 'bytes':
<type 'str'>, 'reduce': <built-in function reduce>, 'intern': <built-
in function intern>, 'issubclass': <built-in function issubclass>,
'Ellipsis': Ellipsis, 'EOFError': <type 'exceptions.EOFError'>,
'locals': <built-in function locals>, 'BufferError': <type 'excep-
tions.BufferError'>, 'slice': <type 'slice'>, 'FloatingPointError':
<type 'exceptions.FloatingPointError'>, 'sum': <built-in function
sum>, 'getattr': <built-in function getattr>, 'abs': <built-in
function abs>, 'exit': Use exit() or Ctrl-D (i.e. EOF) to exit, 'print':
<built-in function print>, 'True': True, 'FutureWarning': <type
'exceptions.FutureWarning'>, 'ImportWarning': <type 'excep-
tions.ImportWarning'>, 'None': None, 'hash': <built-in function
hash>, 'ReferenceError': <type 'exceptions.ReferenceError'>,
'len': <built-in function len>, 'credits': Thanks to CWI, CNRI,
BeOpen.com, Zope Corporation and a cast of thousands for
supporting Python development. See www.python.org for more
information., 'frozenset': <type 'frozenset'>, '__name__':
'__builtin__', 'ord': <built-in function ord>, 'super': <type
'super'>, 'TypeError': <type 'exceptions.TypeError'>, 'license':
See http://www.python.org/2.7/license.html, 'KeyboardInterrupt':
```

Safe imports: rollback after a failed import.

Initially inspired from the RollbackImporter in PyUnit, but it's now much simpler and works better for our needs.

See <http://pyunit.sourceforge.net/notes/reloading.html>

`pymills.utils.notags(str)`

Removes HTML tags from str and returns the new string

`class pymills.utils.MemoryStats(pid=616)`

Bases: object

scale = {'KB': 1024.0, 'MB': 1048576.0}

size

rss

stack

`pymills.utils.memory(since=0.0)`

Return memory usage in bytes.

`pymills.utils.resident(since=0.0)`

Return resident memory usage in bytes.

`pymills.utils.stacksize(since=0.0)`

Return stack size in bytes.

`pymills.utils.Mixin(cls, mixin, last=False)`

`class pymills.utils.Cache(f)`

Bases: object

`pymills.utils.printdict(d, level=0) → None`

Print the given dictionary, d. Recursively print any nested dictionaries found. This function is *_NOT_* a pretty printer, and prints a human-readable form of the given dictionary rather than something that can be re-read by Python.

`pymills.utils.minmax(iter) -> (min, max)`

Consume the iterable iter and calculate and return the min and max of each item.

`pymills.utils.caller(n=1) → str`

Return the name of the calling function. If n is specified, return the n'th function in the stack.

version Module

Version Module

So we only have to maintain version information in one place!

Subpackages

ai Package

deduce Module

`class pymills.ai.deduce.sentence(string)`

adjectives = ['A', 'AN', 'THE', 'ALL', 'EVERY', 'NONE', 'MY', 'YOUR']

```

negatives = ['NO', 'NONE', 'NOT', 'CANNOT', 'CANT', 'WONT', 'DOESNT', 'DONT', 'ISNT', 'ARENT', 'AINT']
verbs = ['IS', 'ARE', 'AM', 'BE', 'WAS', 'WERE', 'BE']
helping_verbs = ['DO', 'DOES', 'CAN', 'WILL', 'HAS', 'HAVE']
contractions = {'DONT': ['DO', 'NOT'], 'ILL': ['I', 'WILL'], 'YOU'RE': ['YOU', 'ARE'], 'ISNT': ['IS', 'NOT'], 'DO'}
next_word()
prev_word()
rewind_words()
next_phrase()
is_question()
class pymills.ai.deduce.fact(statement)

    subj_adj = ''
    subj = ''
    helping_verb = ''
    verb = ''
    orig_verb = ''
    obj_adj = ''
    obj = ''
    question = 0
    negative = 0
    seen = 0
    to_be = ['IS', 'ARE', 'AM', 'BE', 'WAS', 'WERE']
    translate(statement)
    swap_person()
class pymills.ai.deduce.brain

    to_be = ['IS', 'ARE', 'BE', 'WAS', 'WERE']
    learn(newfact)
    describe_subj(subject)
    query(question)
    query_i(q)
pymills.ai.deduce.ui()

```

2.2 TODO

- No immediate plans.

2.3 Road Map

- No immediate future plans at this stage... Feel free to [Create an Issue](#).

2.4 Changes

2.4.1 pymills 3.4.3.dev

-

2.4.2 pymills 3.4.2 (2014-09-16)

- Fixed a minor bug with `pymills.utils.GetFiles()`

2.4.3 pymills 3.4 (2013-11-20)

- Initial Public Release

Note: No previous chagne log was kept.

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `pymills.__init__`, 5
- `pymills.ai.deduce`, 14
- `pymills.datatypes`, 5
- `pymills.dbapi`, 6
- `pymills.emailtools`, 7
- `pymills.mathtools`, 8
- `pymills.misc`, 8
- `pymills.pyodict`, 9
- `pymills.table`, 9
- `pymills.utils`, 11
- `pymills.version`, 14

A

add() (pymills.emailtools.Email method), 7
addPercent() (in module pymills.misc), 8
adjectives (pymills.ai.deduce.sentence attribute), 14
append() (pymills.datatypes.CaselessList method), 6
append() (pymills.table.Table method), 11

B

backMerge() (in module pymills.misc), 8
BaseSession (class in pymills.dbapi), 7
beat() (in module pymills.misc), 8
bottom() (pymills.datatypes.Queue method), 6
brain (class in pymills.ai.deduce), 15
buildAverage() (in module pymills.misc), 8
bytes() (in module pymills.misc), 8

C

Cache (class in pymills.utils), 14
caller() (in module pymills.utils), 14
CaselessDict (class in pymills.datatypes), 6
CaselessList (class in pymills.datatypes), 6
Cell (class in pymills.table), 10
clear() (pymills.datatypes.OrderedDict method), 5
close() (pymills.dbapi.BaseSession method), 7
commit() (pymills.dbapi.BaseSession method), 7
Connection() (in module pymills.dbapi), 6
ConnectionError, 6
contractions (pymills.ai.deduce.sentence attribute), 15
create_connection() (in module pymills.dbapi), 6
cursor() (pymills.dbapi.BaseSession method), 7

D

DatabaseError, 7
describe_subj() (pymills.ai.deduce.brain method), 15
do() (pymills.dbapi.BaseSession method), 7
DriverError, 6
duration() (in module pymills.misc), 8

E

Email (class in pymills.emailtools), 7

empty() (pymills.datatypes.Queue method), 6
empty() (pymills.datatypes.Stack method), 6
Error, 11
execute() (pymills.dbapi.BaseSession method), 7

F

fact (class in pymills.ai.deduce), 15
full() (pymills.datatypes.Queue method), 6

G

get() (pymills.datatypes.CaselessDict method), 6
get() (pymills.datatypes.Queue method), 6
get_mimetype() (in module pymills.emailtools), 8
getFiles() (in module pymills.utils), 12
getHeader() (pymills.table.Cell method), 10
getTotalTime() (in module pymills.misc), 8
getWidth() (pymills.table.Header method), 9
getXY() (pymills.table.Table method), 11

H

has_key() (pymills.datatypes.CaselessDict method), 6
Header (class in pymills.table), 9
helping_verb (pymills.ai.deduce.fact attribute), 15
helping_verbs (pymills.ai.deduce.sentence attribute), 15
hmsToSeconds() (in module pymills.misc), 8

I

index() (pymills.datatypes.OrderedDict method), 5
is_question() (pymills.ai.deduce.sentence method), 15
isReadable() (in module pymills.utils), 12
items() (pymills.datatypes.OrderedDict method), 5
iteritems() (pymills.datatypes.OrderedDict method), 5
iterkeys() (pymills.datatypes.OrderedDict method), 5
itervalues() (pymills.datatypes.OrderedDict method), 5

K

keys() (pymills.datatypes.OrderedDict method), 5

L

learn() (pymills.ai.deduce.brain method), 15

M

mean() (in module pymills.mathtools), 8
memory() (in module pymills.utils), 14
MemoryStats (class in pymills.utils), 14
mimify_file() (in module pymills.emailtools), 8
minmax() (in module pymills.utils), 14
MixIn() (in module pymills.utils), 14
mkpasswd() (in module pymills.utils), 12
mysql_session() (in module pymills.dbapi), 6
MySQLSession (class in pymills.dbapi), 7

N

negative (pymills.ai.deduce.fact attribute), 15
negatives (pymills.ai.deduce.sentence attribute), 14
next() (pymills.dbapi.Records method), 7
next_phrase() (pymills.ai.deduce.sentence method), 15
next_word() (pymills.ai.deduce.sentence method), 15
notags() (in module pymills.utils), 14
Null (class in pymills.datatypes), 6

O

obj (pymills.ai.deduce.fact attribute), 15
obj_adj (pymills.ai.deduce.fact attribute), 15
odict (class in pymills.pyodict), 9
ORACLE_ARRAYSIZE (pymills.dbapi.OracleSession attribute), 7
oracle_session() (in module pymills.dbapi), 6
OracleSession (class in pymills.dbapi), 7
OrderedDict (class in pymills.datatypes), 5
orig_verb (pymills.ai.deduce.fact attribute), 15

P

peek() (pymills.datatypes.Queue method), 6
peek() (pymills.datatypes.Stack method), 6
pop() (pymills.datatypes.Queue method), 6
pop() (pymills.datatypes.Stack method), 6
popitem() (pymills.datatypes.OrderedDict method), 5
prev_word() (pymills.ai.deduce.sentence method), 15
printdict() (in module pymills.utils), 14
push() (pymills.datatypes.Queue method), 6
push() (pymills.datatypes.Stack method), 6
pymills.__init__ (module), 5
pymills.ai.deduce (module), 14
pymills.datatypes (module), 5
pymills.dbapi (module), 6
pymills.emailtools (module), 7
pymills.mathtools (module), 8
pymills.misc (module), 8
pymills.pyodict (module), 9
pymills.table (module), 9
pymills.utils (module), 11
pymills.version (module), 14

Q

query() (pymills.ai.deduce.brain method), 15
query_i() (pymills.ai.deduce.brain method), 15
question (pymills.ai.deduce.fact attribute), 15
Queue (class in pymills.datatypes), 6

R

Record (class in pymills.dbapi), 7
Records (class in pymills.dbapi), 7
refresh() (pymills.table.Cell method), 10
refresh() (pymills.table.Header method), 9
refresh() (pymills.table.Row method), 10
refresh() (pymills.table.Table method), 11
remove() (pymills.datatypes.CaselessList method), 6
resident() (in module pymills.utils), 14
rewind_words() (pymills.ai.deduce.sentence method), 15
rollback() (pymills.dbapi.BaseSession method), 7
Row (class in pymills.table), 9
rss (pymills.utils.MemoryStats attribute), 14

S

safe__import__() (in module pymills.utils), 12
scale (pymills.utils.MemoryStats attribute), 14
seen (pymills.ai.deduce.fact attribute), 15
send() (pymills.emailtools.Email method), 7
send_email() (in module pymills.emailtools), 8
sendEmail() (in module pymills.emailtools), 7
sentence (class in pymills.ai.deduce), 14
set() (pymills.utils.State method), 11
setdefault() (pymills.datatypes.OrderedDict method), 5
size (pymills.utils.MemoryStats attribute), 14
size() (pymills.datatypes.Queue method), 6
sqlite_session() (in module pymills.dbapi), 6
SQLiteSession (class in pymills.dbapi), 7
Stack (class in pymills.datatypes), 5
stack (pymills.utils.MemoryStats attribute), 14
stacksize() (in module pymills.utils), 14
State (class in pymills.utils), 11
std() (in module pymills.mathtools), 8
strToBool() (in module pymills.misc), 8
subj (pymills.ai.deduce.fact attribute), 15
subj_adj (pymills.ai.deduce.fact attribute), 15
subPercent() (in module pymills.misc), 8
swap_person() (pymills.ai.deduce.fact method), 15

T

Table (class in pymills.table), 10
test() (in module pymills.table), 11
to_be (pymills.ai.deduce.brain attribute), 15
to_be (pymills.ai.deduce.fact attribute), 15
toHTML() (pymills.table.Cell method), 10
toHTML() (pymills.table.Header method), 9
toHTML() (pymills.table.Row method), 10

`toHTML()` (pymills.table.Table method), [11](#)
`top()` (pymills.datatypes.Queue method), [6](#)
`translate()` (pymills.ai.deduce.fact method), [15](#)

U

`ui()` (in module pymills.ai.deduce), [15](#)
`update()` (pymills.datatypes.OrderedDict method), [5](#)

V

`validateEmail()` (in module pymills.utils), [12](#)
`values()` (pymills.datatypes.OrderedDict method), [5](#)
`verb` (pymills.ai.deduce.fact attribute), [15](#)
`verbs` (pymills.ai.deduce.sentence attribute), [15](#)